Manual de Programador

Football Life



Tomás Silva

Índice

INTRODUÇÃO 1

BASE DE DADOS 2

*CURRENT USER* 3

LOGIN.CS 4

QUERYS E NONQUERYS 5

PAGINAÇÃO DE DADOS DA BASE DE DADOS 5

*USER CONTROL* – “JOGO” 6

JOGOS - IMAGENS 7

GUARDAR FATURA COTA – *TXT FILES* 7

RESTRIÇÕES DAS TEXTBOXS NUMÉRICAS 8

PEQUENAS NOTAS 8

## INTRODUÇÃO

Este programa foi desenvolvido em C#, no Visual Studio 2019, em Windows Forms e tem uma base de dados criada no Microsoft SQL Server Management Studio 18.

O utilizador também deverá ter uma capacidade mínima de 2GB de RAM na sua máquina e um sistema operativo Windows para o programa ter capacidade para exercer todas as suas funções.

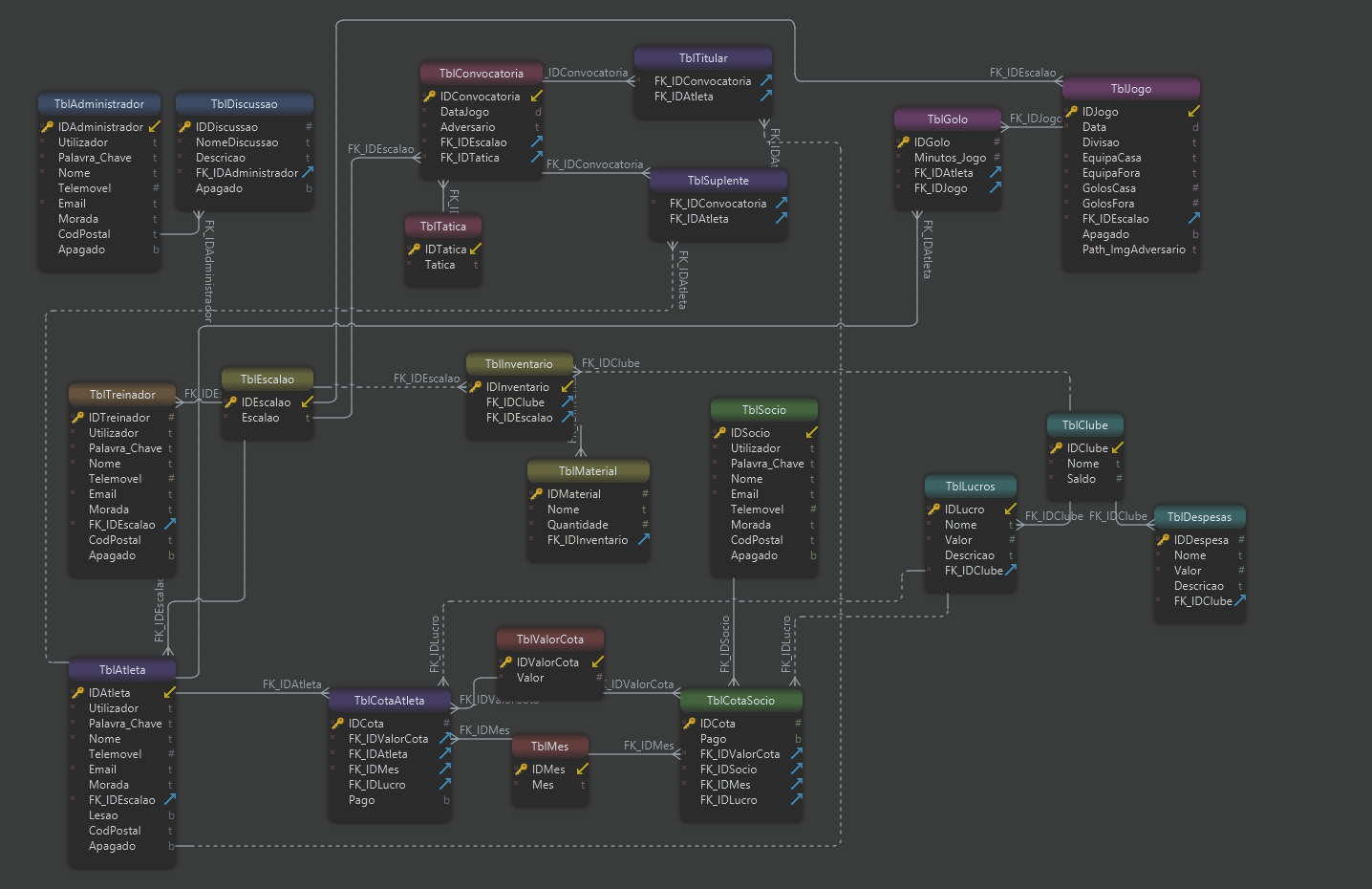
Este programa é muito baseado em ler, modificar e eliminar dados da base de dados, não tendo muitas funções sem o uso da base de dados.

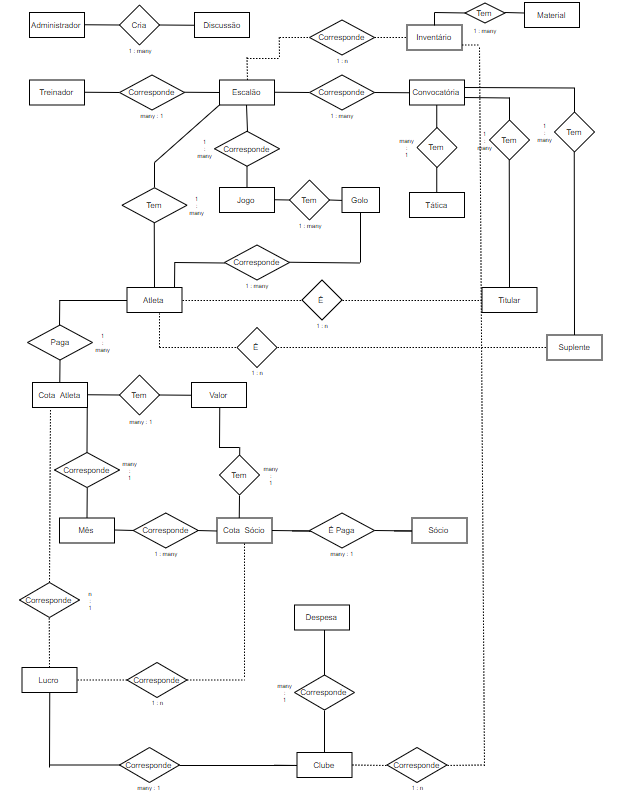
Neste manual, vai ser explicado como a base de dados funciona e as principais e mais difíceis partes de código de “Football Life”, para que o programador possa percebê-lo e mais tarde alterá-lo, caso necessite.

## BASE DE DADOS

A base de dados do programa tem por volta de 26 tabelas sendo 4 delas de utilizadores, ou seja, a tabela “TblAdministrador”, a tabela “TblTreinador”, a tabela “TblAtletas” e a tabela “TblSocio”. Isto para que a relação com as outras tabelas fosse mais fiável e compreensível.

A esquema da base de dados é esta imagem abaixo e segue o seu diagrama de relação-entidade:



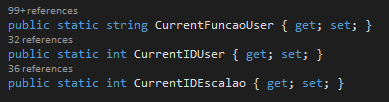


## *CURRENT USER*

Para guardar valores sobre o *current user*, foram utilizadas propriedades situadas no “Program.cs”. O seu nome e as respetivas funções são as seguintes:

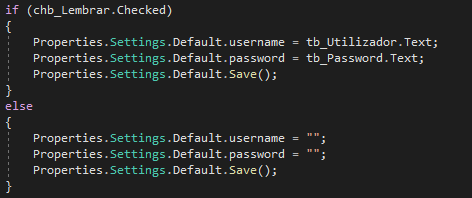
* **CurrentFuncaoUser –** Indica qual é a função do current user. Os seus valores são guardados no “Login.cs” e são valores únicos, atribuídos pelo programador (“Admin”, “Treinador”, “Atleta” e “Socio”);

* **CurrentIDUser –** Indica qual é o ID do current user. O seu valor é guardado no “Login.cs” e corresponde a um valor da base de dados;
* **CurrentIDEscalao –** Indica qual o ID do escalão do current user, caso tiver (se for treinador ou atleta). O seu valor é guardado no “Login.cs” e corresponde a um valor da base de dados.



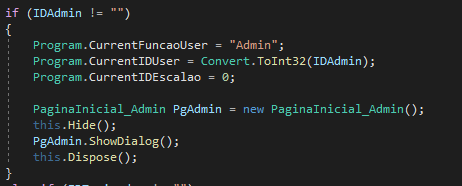
## LOGIN.CS

É no “Login.cs” que realmente começa o código mais complicado.   
Para começar o utilizador tem a opção de guardar a sua password e para isso usei as Propriedades do programa e inseri duas variáveis a para o nome do utilizador e outra para a password. Foi utilizada esta forma, porque mesmo depois do programa ser fechado os valores ficarão guardados.

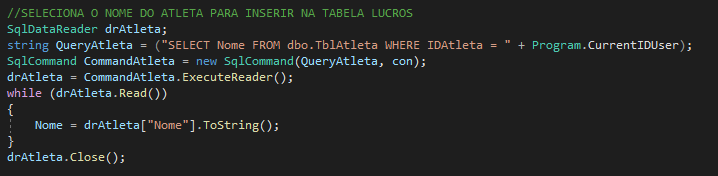


Para, então, fazer o login, foram utilizados DataReaders, com querys a pedir utilizadores em todas as tabelas com o nome e a password correspondestes. Como os utilizadores não podem tem utilizadores iguais, apenas umas das tabelas (TblAdministrador, TblTreinador, TblAtleta e TblSocio) dará valores.

Quando feito o Login, então as propriedades faladas anteriormente, deixam de estar vazias e tem um valor para o funcionamento do resto do programa.

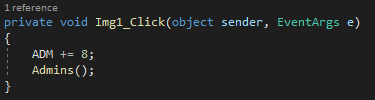


## QUERYS E NONQUERYS

 Para manusear com a base de dados, seja com o uso de querys (Selets) ou nonquerys (Inserts, Updates e Deletes) foram usados “SqlCommands”. Para isto é necessário o uso da biblioteca System.Data.SqlClient.

## PAGINAÇÃO DE DADOS DA BASE DE DADOS

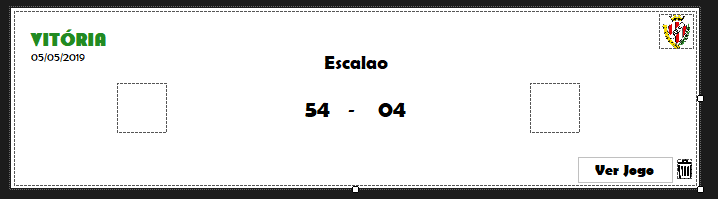
Como alguns forms têm inúmeros dados a serem carregados, adicionei a função paginação para que mesmos dados carregassem ao mesmo tempo e esta função pode ser encontrada nos forms: Jogos, PaginaInicial\_Admin, PaginaInicial\_Treinador, PaginaInicial\_Atleta, PaginaInicial\_Socio, Utilizadores e Financiamento. Isto está representado na seguinte imagem e tem um código que adiciona mais um certo valor as quantas vezes que o utilizador clicar na imagem.



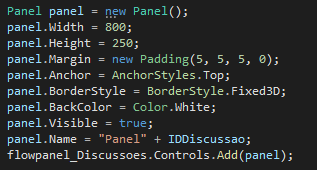


## *USER CONTROL* – “JOGO”

Um *user control* foi usado para jogos. Neste caso, o *user control* é o resultado de um jogo e cada vez que um jogo é lido da base de dados o *user control* “Jogo” é adicionado ao *flowlayoutpanel* dos jogos, pois fornece maior flexibilidade e maior facilidade para a reutilização.

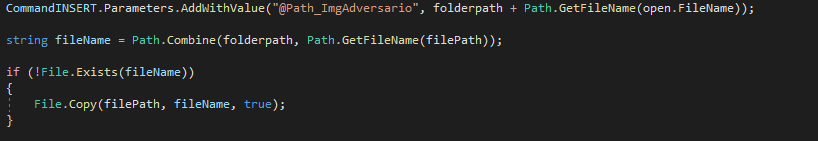


Para além do *user control*, foi usado também algo parecido, mas criado em código, criando assim *labels*, *buttons*, entre outras coisas em código, tal como na imagem.

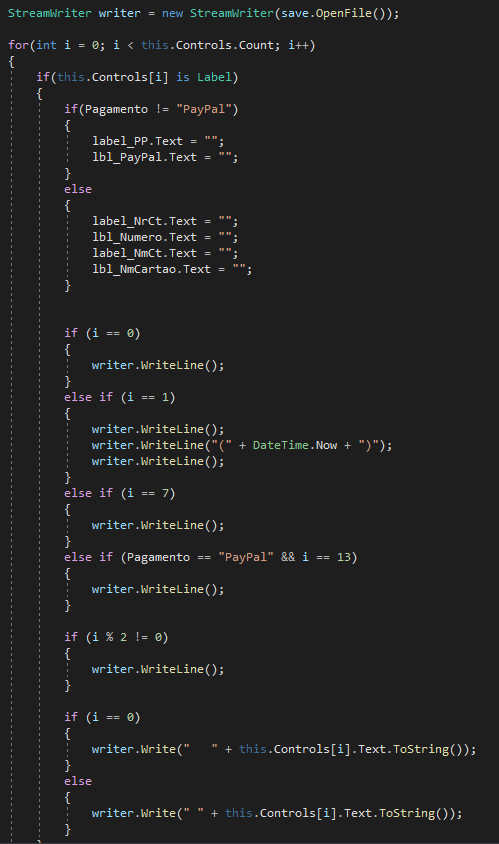


## JOGOS - IMAGENS

Todos os jogos têm uma imagem de um logo adversário e, para isto, foi necessário adicionar o caminho da imagem selecionada à base de dados, para isto o programa copia a imagem para uma pasta dentro do programa e guarda o caminho para a base de dados. Para isto é necessário usar a biblioteca System.IO.

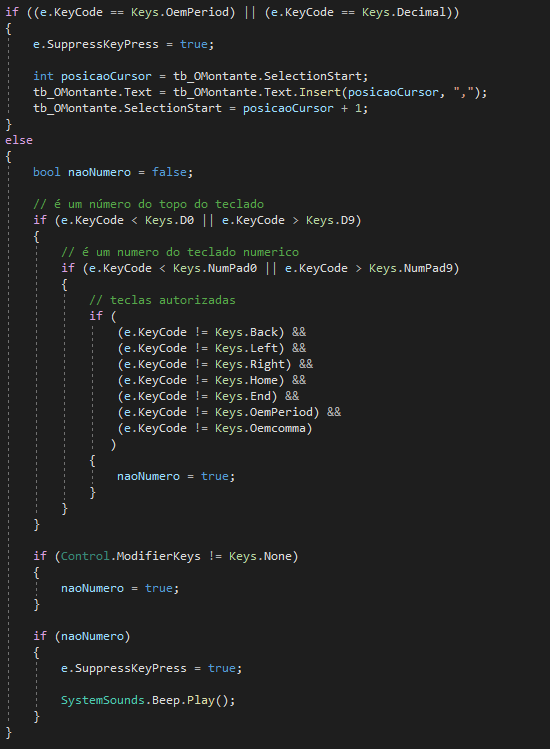


## GUARDAR FATURA COTA – *TXT FILES*



Quando o utilizador paga a sua cota tem direito a salvar uma fatura que lhe é dada. Apesar de não ser muito seguro porque o utilizador pode alterar o ficheiro, esta e gravada como um ficheiro de texto. Para isto é usado um “*for*” para o ficheiro verificar tudo o que está dentro do form “Fatura Cota” e salvar todas essas informações no ficheiro de texto. Para isto é necessário usar a biblioteca System.IO.

## RESTRIÇÕES DAS TEXTBOXS NUMÉRICAS



Para restringir o utilizador à utilização de números em certas caixas de texto, foi utilizado a biblioteca System.Media para caso do utilizador estar a introduzir um valor não número ele ser notificado.

Em algumas partes deste código também não só foram restringidas as caixas de texto, mas também foram substituídos o ponto por virgula ou o ponto por dois pontos.

## PEQUENAS NOTAS

O código para as cotas ainda não está a funcionar corretamente, pois o PayPal ou os Multibancos não são verificados, nem valores são retirados dos mesmos, ou seja, o código está ainda para teste.